Subject: Re: WaitForMultipleObjects() analog?
Posted by hojtsy on Thu, 29 May 2008 13:12:30 GMT
View Forum Message <> Reply to Message

Mindtraveller,

AFAIK unix can only block on multiple conditions inside "select".
Anything else is emulation. The code you attached can indeed wait on multiple Semaphores, but not on file descriptors which you also need. That is the more interesting part I think.

Implementation alternative A:

You can block on a single Semaphore or Monitor, which protects a message queue. In this case any of the events you list should be an other thread putting a message to the queue, which the worker thread awakes to process. The direct IO communication (the select call) should be done on a different thread which puts incoming messages to the message queue and signals the Semaphore/Monitor.
In this case the problem arises how do you cancel the IO thread which is blocking inside select when you want to shutdown? You have three options: either call shutdown on the file description it is waiting for, or have it select on a pipe too which only gets written to when a shutdown occurs, or have the thread automatically wake up every 0,1 sec from the select and poll for the shutdown flag. In all cases the IO thread gets waken up from select and can exit. The worker thread could just wake up on a shutdown message arriving in it's message queue.

Implementation alternative B:

You can block inside a select which waits for the IO port and multiple pipes, each of which communicates different events. Timeout is automatically handled by select, shutdown and queue events could be other threads writing some bytes to the dedicated pipes, which also wakes up the thread doing the select.

I think solution A is better because it separates the low level pipe hacking from the main logic of the application, needs less pipes, and also makes the main part portable without preprocessor guards.